

# ATLAS ANSI/ISO C BLAS API REFERENCE

ROUTINE	(ARGUMENTS)	DESCRIPTION	PREFIXES
<b>Level 1 BLAS</b>			
void cblas_◇rotg	(TYPE *a, TYPE *b, TYPE *c, TYPE *s)	Generate plane rotation	S, D
void cblas_◇rotg	(TYPE *a, TYPE *b, TYPE *c, TYPE *s)	Generate plane rotation	C, Z
void cblas_◇rotmg	(TYPE *d1, TYPE *d2, TYPE *b1, SCALAR b2, TYPE *P)	Generate modified plane rotation	S,D
void cblas_◇rot	(const int N, TYPE *X, const int incX, TYPE *Y, const int incY, SCALAR c, SCALAR s)	Apply plane rotation	S,D
void cblas_◇rot	(const int N, TYPE *X, const int incX, TYPE *Y, const int incY, const UTYPE c, const UTYPE s)	Apply plane rotation	CS,ZD
void cblas_◇rotm	(const int N, TYPE *X, const int incX, TYPE *Y, const int incY, SCALAR c, TYPE *P)	Apply modified plane rotation	S,D
void cblas_◇scal	(const int N, SCALAR alpha, TYPE *X, const int incX)	$x \leftarrow y$	S,D,C,Z,CS,ZD
void cblas_◇copy	(const int N, const TYPE *X, const int incX, TYPE *Y, const int incY)	$y \leftarrow x$	S,D,C,Z
void cblas_◇axpy	(const int N, SCALAR alpha, const TYPE *X, const int incX, TYPE *Y, const int incY)	$y \leftarrow \alpha x + y$	S,D,C,Z
TYPE cblas_◇dot	(const int N, const TYPE *X, const int incX, const TYPE *Y, const int incY)	$cblas\_dot \leftarrow x^T y$	S,D,DS
void cblas_◇dotu_sub	(const int N, const TYPE *X, const int incX, const TYPE *Y, const int incY, TYPE *dotu)	$dotu \leftarrow x^T y$	C,Z
void cblas_◇dotc_sub	(const int N, const TYPE *X, const int incX, const TYPE *Y, const int incY, TYPE *dotc)	$dotc \leftarrow x^H y$	C,Z
float cblas_sdsdot	(const int N, const float alpha, const float *X, const int incX, const float *Y, const int incY)	$dot \leftarrow \alpha + x^T y$	SDS
UTYPE cblas_◇nrm2	(const int N, const TYPE *X, const int incX)	$cblas\_nrm2 \leftarrow \ x\ _2$	S,D,SC,DZ
UTYPE cblas_◇asum	(const int N, const TYPE *X, const int incX)	$cblas\_asum \leftarrow \ re(x)\ _1 + \ im(x)\ _1$	S,D,SC,DZ
CBLAS_INDEX cblas_i◇amax	(const int N, const TYPE *X, const int incX)	$amax \leftarrow 1^{st} k \ni  re(x_k)  +  im(x_k) $	S,D,C,Z
<b>Level 3 BLAS</b>			
void cblas_◇gemm	( const enum CBLAS_ORDER Order, const enum CBLAS_TRANSPOSE TransA, const enum CBLAS_TRANSPOSE TransB, const int M, const int N, const int K, const SCALAR alpha, const TYPE *A, const int lda, const TYPE *B, const int ldb, const SCALAR beta, TYPE *C, const int ldc )	$C \leftarrow \alpha op(A) op(B) + \beta C$ , $op(X) = X, X^T, X^H, C - m \times n$	S,D,C,Z
void cblas_◇symm	( const enum CBLAS_ORDER Order, const enum CBLAS_SIDE Side, const enum CBLAS_UPLO Uplo, const int M, const int N, SCALAR alpha, const TYPE *A, const int lda, const TYPE *B, const int ldb, SCALAR beta, TYPE *C, const int ldc )	$C \leftarrow \alpha AB + \beta C, C \leftarrow \alpha BA + \beta C, C - m \times n, A = A^T$	S,D,C,Z
void cblas_◇hemm	( const enum CBLAS_ORDER Order, const enum CBLAS_SIDE Side, const enum CBLAS_UPLO Uplo, const int M, const int N, const void *alpha, const void *A, const int lda, const void *B, const int ldb, const void *beta, void *C, const int ldc )	$C \leftarrow \alpha AB + \beta C, C \leftarrow \alpha BA + \beta C, C - m \times n, A = A^H$	C,Z
void cblas_◇syrk	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const enum CBLAS_TRANSPOSE Trans, const int N, const int K, SCALAR alpha, const TYPE *A, const int lda, SCALAR beta, TYPE *C, const int ldc )	$C \leftarrow \alpha AA^T + \beta C, C \leftarrow \alpha A^T A + \beta C, C - n \times n$	S,D,C,Z
void cblas_◇herk	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const enum CBLAS_TRANSPOSE Trans, const int N, const int K, const UTYPE alpha, const void *A, const int lda, const UTYPE beta, void *C, const int ldc )	$C \leftarrow \alpha AA^H + \beta C, C \leftarrow \alpha A^H A + \beta C, C - n \times n$	C,Z
void cblas_◇syr2k	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const enum CBLAS_TRANSPOSE Trans, const int N, const int K, SCALAR alpha, const TYPE *A, const int lda, const TYPE *B, const int ldb, SCALAR beta, TYPE *C, const int ldc )	$C \leftarrow \alpha AB^T + \bar{\alpha} BA^T + \beta C, C \leftarrow \alpha A^T B + \bar{\alpha} B^T A + \beta C, C - n \times n$	S,D,C,Z
void cblas_◇her2k	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const enum CBLAS_TRANSPOSE Trans, const int N, const int K, const void *alpha, const void *A, const int lda, const void *B, const int ldb, const UTYPE beta, void *C, const int ldc )	$C \leftarrow \alpha AB^H + \bar{\alpha} BA^H + \beta C, C \leftarrow \alpha A^H B + \bar{\alpha} B^H A + \beta C, C - n \times n$	C,Z
void cblas_◇trmm	( const enum CBLAS_ORDER Order, const enum CBLAS_SIDE Side, const enum CBLAS_UPLO Uplo, const enum CBLAS_TRANSPOSE TransA, const enum CBLAS_DIAG Diag, const int M, const int N, SCALAR alpha, const TYPE *A, const int lda, TYPE *B, const int ldb )	$B \leftarrow \alpha op(A) B, B \leftarrow \alpha B op(A), op(A) = A, A^T, A^H, B - m \times n$	S,D,C,Z
void cblas_◇trsm	( const enum CBLAS_ORDER Order, const enum CBLAS_SIDE Side, const enum CBLAS_UPLO Uplo, const enum CBLAS_TRANSPOSE TransA, const enum CBLAS_DIAG Diag, const int M, const int N, SCALAR alpha, const TYPE *A, const int lda, TYPE *B, const int ldb )	$B \leftarrow \alpha op(A^{-1}) B, B \leftarrow \alpha B op(A^{-1}), op(A) = A, A^T, A^H, B - m \times n$	S,D,C,Z

## NOTES:

- Routines in *italics* are not mandated by the BLAS standard.
- Calling routines should include the standard header file, `cblas.h`.
- More information available at <http://math-atlas.sourceforge.net/>.

## PREFIX RELATED DEFINITIONS :

◇is	Data operated	TYPE	UTYPE	SCALAR
s	single precision real	float	float	const float
d	double precision real	double	double	const double
c	single precision complex	void	float	const void*
z	double precision complex	void	double	const void*

# ATLAS ANSI/ISO C BLAS API REFERENCE

ROUTINE	(ARGUMENTS)	DESCRIPTION	PREFIXES
<b>Level 2 BLAS</b>			
void cblas_◇gemv	( const enum CBLAS_ORDER Order, const enum CBLAS_TRANSPOSE TransA, const int M, const int N, SCALAR alpha, const TYPE *A, const int lda, const TYPE *X, const int incX, SCALAR beta, TYPE *Y, const int incY )	$y \leftarrow \alpha Ax + \beta y, \quad y \leftarrow \alpha A^T x + \beta y,$ $y \leftarrow \alpha A^H x + \beta y, A - m \times n$	S,D,C,Z
void cblas_◇gbmv	( const enum CBLAS_ORDER Order, const enum CBLAS_TRANSPOSE TransA, const int M, const int N, const int KL, const int KU, SCALAR alpha, const TYPE *A, const int lda, const TYPE *X, const int incX, SCALAR beta, TYPE *Y, const int incY )	$y \leftarrow \alpha Ax + \beta y, \quad y \leftarrow \alpha A^T x + \beta y,$ $y \leftarrow \alpha A^H x + \beta y, A - m \times n$	S,D,C,Z
void cblas_◇hemv	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const int N, const void *alpha, const void *A, const int lda, const void *X, const int incX, const void *beta, void *Y, const int incY )	$y \leftarrow \alpha Ax + \beta y$	C,Z
void cblas_◇hbmV	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const int N, const int K, const void *alpha, const void *A, const int lda, const void *X, const int incX, const void *beta, void *Y, const int incY )	$y \leftarrow \alpha Ax + \beta y$	C,Z
void cblas_◇hpmv	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const int N, const void *alpha, const void *Ap, const void *X, const int incX, const void *beta, void *Y, const int incY )	$y \leftarrow \alpha Ax + \beta y$	C,Z
void cblas_◇symv	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const int N, SCALAR alpha, const TYPE *A, const int lda, const TYPE *X, const int incX, SCALAR beta, TYPE *Y, const int incY )	$y \leftarrow \alpha Ax + \beta y$	S,D
void cblas_◇sbmv	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const int N, const int K, SCALAR alpha, const TYPE *A, const int lda, const TYPE *X, const int incX, SCALAR beta, TYPE *Y, const int incY )	$y \leftarrow \alpha Ax + \beta y$	S,D
void cblas_◇spmv	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const int N, SCALAR alpha, const TYPE *Ap, const TYPE *X, const int incX, SCALAR beta, TYPE *Y, const int incY )	$y \leftarrow \alpha Ax + \beta y$	S,D
void cblas_◇trmv	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const enum CBLAS_TRANSPOSE TransA, const enum CBLAS_DIAG Diag, const int N, const TYPE *A, const int lda, TYPE *X, const int incX )	$x \leftarrow Ax, x \leftarrow A^T x, x \leftarrow A^H x$	S,D,C,Z
void cblas_◇tbmv	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const enum CBLAS_TRANSPOSE TransA, const enum CBLAS_DIAG Diag, const int N, const int K, const TYPE *A, const int lda, TYPE *X, const int incX )	$x \leftarrow Ax, x \leftarrow A^T x, x \leftarrow A^H x$	S,D,C,Z
void cblas_◇tpmv	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const enum CBLAS_TRANSPOSE TransA, const enum CBLAS_DIAG Diag, const int N, const TYPE *Ap, TYPE *X, const int incX )	$x \leftarrow Ax, x \leftarrow A^T x, x \leftarrow A^H x$	S,D,C,Z
void cblas_◇trsv	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const enum CBLAS_TRANSPOSE TransA, const enum CBLAS_DIAG Diag, const int N, const TYPE *A, const int lda, TYPE *X, const int incX )	$x \leftarrow A^{-1}x, x \leftarrow A^{-T}x, x \leftarrow A^{-H}x$	S,D,C,Z
void cblas_◇tbsv	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const enum CBLAS_TRANSPOSE TransA, const enum CBLAS_DIAG Diag, const int N, const int K, const TYPE *A, const int lda, TYPE *X, const int incX )	$x \leftarrow A^{-1}x, x \leftarrow A^{-T}x, x \leftarrow A^{-H}x$	S,D,C,Z
void cblas_◇tpsv	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const enum CBLAS_TRANSPOSE TransA, const enum CBLAS_DIAG Diag, const int N, const TYPE *Ap, TYPE *X, const int incX )	$x \leftarrow A^{-1}x, x \leftarrow A^{-T}x, x \leftarrow A^{-H}x$	S,D,C,Z
void cblas_◇ger	( const enum CBLAS_ORDER Order, const int M, const int N, SCALAR alpha, const TYPE *X, const int incX, const TYPE *Y, const int incY, TYPE *A, const int lda )	$A \leftarrow \alpha xy^T + A, A - m \times n$	S,D
void cblas_◇geru	( const enum CBLAS_ORDER Order, const int M, const int N, const void *alpha, const void *X, const int incX, const void *Y, const int incY, void *A, const int lda )	$A \leftarrow \alpha xy^T + A, A - m \times n$	C,Z
void cblas_◇gerc	( const enum CBLAS_ORDER Order, const int M, const int N, const void *alpha, const void *X, const int incX, const void *Y, const int incY, void *A, const int lda )	$A \leftarrow \alpha xy^H + A, A - m \times n$	C,Z
void cblas_◇her	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const int N, const UTYPE alpha, const void *X, const int incX, void *A, const int lda )	$A \leftarrow \alpha xx^H + A$	C,Z
void cblas_◇hpr	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const int N, const UTYPE alpha, const void *X, const int incX, void *A )	$A \leftarrow \alpha xx^H + A$	C,Z
void cblas_◇her2	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const int N, const void *alpha, const void *X, const int incX, const void *Y, const int incY, void *A, const int lda )	$A \leftarrow \alpha xy^H + y(\alpha x)^H + A$	C,Z
void cblas_◇hpr2	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const int N, const void *alpha, const void *X, const int incX, const void *Y, const int incY, void *Ap )	$A \leftarrow \alpha xy^H + y(\alpha x)^H + A$	C,Z
void cblas_◇syr	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const int N, SCALAR alpha, const TYPE *X, const int incX, TYPE *A, const int lda )	$A \leftarrow \alpha xx^T + A$	S,D
void cblas_◇spr	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const int N, SCALAR alpha, const TYPE *X, const int incX, TYPE *Ap )	$A \leftarrow \alpha xx^T + A$	S,D
void cblas_◇syr2	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const int N, SCALAR alpha, const TYPE *X, const int incX, const TYPE *Y, const int incY, TYPE *A, const int lda )	$A \leftarrow \alpha xy^T + \alpha yx^T + A$	S,D
void cblas_◇spr2	( const enum CBLAS_ORDER Order, const enum CBLAS_UPLO Uplo, const int N, SCALAR alpha, const TYPE *X, const int incX, const TYPE *Y, const int incY, TYPE *A )	$A \leftarrow \alpha xy^T + \alpha yx^T + A$	S,D